

Toni Svenström

Rahapelien grafiikan suunnittelu ja toteutus

Case Kulta-Jaska ja Unelmaloma

Metropolia Ammattikorkeakoulu
Medianomi (AMK)
Viestinnän koulutusohjelma
Opinnäytetyö
6.6.2011

Tekijä(t) Otsikko Sivumäärä Aika	Toni Svenström Rahapeli grafiikan suunnittelu ja toteutus: Case Kulta-Jaska ja Unelmaloma 37 sivua 6.6.2011
Tutkinto	Medianomi (AMK)
Koulutusohjelma	Viestinnän koulutusohjelma
Suuntautumisvaihtoehto	3D-visualisointi
Ohjaaja(t)	Jaro Lehtonen
<p>Opinnäytetyö käsittelee peligrafiikan luomista Raha-automaattiyhdistyksen kahteen moni- linjaiseen hedelmäpeliin, Kulta-Jaskaan ja Unelmalomaan.</p> <p>Pelit toteutettiin hyvin samankaltaisessa työympäristössä ja samanlaisilla tekniikoilla. Joi- tain eroja projektien välillä kuitenkin oli ja niillä oli vaikutuksia lopputulokseen.</p> <p>Tekstin ensimmäinen osio käsittelee työympäristöä, grafiikan suunnitteluvaihetta ja suunnittelu- n päälinjauksia. Rahapeli suunnittelussa päteviä erityispiirteitä pohditaan grafiikan tekemisen kannalta. Samaten käsitellään Raha-automaattiyhdistyksen erityisasemaa ja peliautomaattien laitteiston ja toimintaympäristön poikkeuksellisuutta.</p> <p>Toinen osio käsittelee pelien teknistä toteutusta, syitä työkalujen ja tekotapojen valintaan ja näistä valinnoista johtuvia erikoistilanteita. 3D-grafiikan hyödyntämistä 2D-grafiikan toteuttamisessa käsitellään ensin yleisesti ja sen jälkeen esitellään käytännön työtapoja 3D Studio Maxia ja Photoshopia käyttäen. Työtapojen esittely ei mene yksityiskohtiin, vaan selittää metodit kyseiseen erikoistilanteeseen sopivasti, näin ollen tekstin ymmärtäminen vaatii käsiteltävien ohjelmien perustaitojen hallinnan. Reaaliaikaisen 3D-grafiikan ongelma- kohtia, kuten läpikuultavien pintojen esittämistä käydään läpi kaikkein tarkimmin. Myös tiedostoformaattien valintoja tekstuureja varten käsitellään.</p> <p>Lopussa opinnäytetyön tekemisestä saadut hyödyt ja case-esimerkit arvioidaan yleisesti. Samalla arvioidaan, miten työtavat tulevat ehkä muuttumaan tulevaisuudessa ja millaista henkilökohtaista kehittämistä työ vaatisi.</p>	
Avainsanat	3D 2D reaaliaika grafiikka peli casino rahapeli suunnittelu scrum

Author(s) Title	Toni Svenström Creating graphics for multireel videoslot games
Number of Pages Date	37 pages 6 June 2011
Degree	Bachelor of Culture and Arts
Degree Programme	Degree programme in Media
Specialisation option	3D-Visualisation
Instructor(s)	Jaro Lehtonen
<p>This thesis describes the design and development process of graphics for two multireel videoslot games, Kulta-Jaska and Unelmaloma that were published and developed by Raha-automaattiyhdistys.</p> <p>Both games were made using similar techniques and in a similar work environment. However, slight differences in both projects ended up affecting the outcome in interesting ways.</p> <p>The first part of the text describes the working environment for both games. Kulta-Jaska was developed with professional, but rather informal project management. The project management type had changed to Scrum when Unelmaloma was made. The use of Scrum changed the attitude towards goal specifications and task timing. The differences in the two ways of project management were compared and evaluated from a graphic artist's perspective. Both games were developed in small teams with slight differences in team structures.</p> <p>The design phase of the graphics is treated in the second section of the text. Videoslot games and casino-games in general stress unique points on graphic design. The games are usually rather static and share a very similar game mechanic with each other. Players come from more varied backgrounds than in traditional videogames and the platform is technically different from most gaming devices. Clarity is the most important thing to remember in the graphic design, but it must be balanced with enough fun details to keep players interested. Animation, timing and rhythm must also be considered in the graphics to make the game feel whole.</p> <p>The third section of the thesis describes some design decisions from a technical point of view. The capabilities and limitations of the platform and graphics engine were considered in a situation where 3D-graphics must be utilized to produce cartoony 2D-graphics in the form of cutout-type animations. The techniques to produce the graphics in 3D Studio Max and Photoshop are explained, but have an emphasis on the specific problems of real-time 3D-graphics engines and texture formats. Other parts of the work are described in a manner that expects the reader to be versed in usual 3D-modelling, animation and Photoshop techniques.</p> <p>Lastly in the text, the working methods and finished products are evaluated as a whole and the needs of future projects are approximated from a personal, technical and work-environment point of view.</p>	
Keywords	3D 2D real-time graphics game casino multireel videoslot design scrum

Sisällys

1	Johdanto	1
2	Huomioitavaa tekijänoikeuksista	2
3	Case-esimerkkien esittely	3
3.1	Kulta-Jaska	3
3.2	Unelmaloma	4
4	Työympäristö ja tiimityöskentely	5
4.1	Pelien projektihallinta	5
4.2	Muut työskentelyyn vaikuttavat tekijät	6
5	Toteutusvaiheet	8
5.1	Suunnittelu	8
5.2	Konseptitaide	9
5.3	Placeholder-grafiikka	10
5.4	Grafiikan suunnittelun periaatteita	11
5.4.1	Kilpailijoiden huomioonotto	11
5.4.2	Automaattien laitteisto ja ympäristö	11
5.4.3	Asiakaskunta	12
5.4.4	Pelien selkeys ja elementtien tärkeysjärjestys	12
5.4.5	Tarinankerronta staattisessa peliympäristössä	13
5.4.6	Ajoitus, rytmitys ja huomion kohdistaminen	14
5.4.7	Graafisten elementtien tärkeysjärjestys toteutuksen kannalta	15
5.4.8	Käyttökelpoisuus pelin markkinointiin	15
6	Grafiikan tekninen toteutus	16
6.1	Peligraafikon suhde tekniikkaan ja peliohjelmaan	16
6.2	Aihepiirin termit	16
6.3	Kaksiulotteisuuden hyvät puolet	19
6.4	3D-grafiikan käytön perustelut	20
6.5	Läpinäkyvyydet reaaliaikaisessa 3D-grafiikassa	23
6.6	2D-hahmon tekeminen Photoshopissa ja 3D Studio Maxissa	24
6.6.1	Hahmojen maalaus ja tekstuurien kokoaminen	24

6.6.2	Mallinnus, riggaus ja animointi	25
6.6.3	Photoshop ja alpha-kanavan tallennus	27
6.6.4	DDS-tiedostot	31
6.6.5	Tekstuurien koko	32
6.7	Työtiedostojen organisointi	32
7	Yhteenveto ja lopputuloksen arviointia	35
7.1	Valmiiden tuotteiden arviointi	35
7.2	Opinnäytetyöstä saavutettu hyöty	35
7.3	Työskentely tulevaisuudessa	36
	Kirjalähteet	37
	Verkkolähteet	37
	Kuvalähteet	37

1 Johdanto

Rahapeleissä grafiikalla on hieman eri painotukset kuin tavallisissa haasteeseen tai tarinankerrontaan painottavissa peleissä. Rahapeleissä pelilliset tapahtumat toistuvat usein pelistä toiseen hyvin samanlaisina ja peliympäristö on vaikkapa puzzlepeleihin verrattavissa staattisuudessaan. Grafiikka on kuitenkin aina videopeleissä yksi vahvimpia elämyksiä ja houkuttelee pelaamaan ja tutkimaan lisää.

Yksi suosituimmista rahapeliformaateista on monilinjaiset videohedelmäpelit. Monilinjaiset ovat hedelmäpelien perintöä, mutta niihin on lisätty enemmän ominaisuuksia kuten voittolinjojen tai kiekkojen määrää ja erilaisia bonuspelejä. Monilinjaisiin on helppoa rakentaa päälle teema, joka erottaa sen muista vastaavista. Monilinjaisten tekemistä voisi verrata flippereiden tekemiseen, sillä pelaaja ohjaa peliä lopulta aina identtisellä tavalla vaikka peliin onkin lisätty pinnallista variaatiota.

Opinnäytetyön case-esimerkit, Kulta-Jaska ja Unelmaloma ovat monilinjaisia videohedelmäpelejä. Pelit ovat hyvin samankaltaisia keskenään, mutta kumpikin toteutettiin hieman eri työtavoilla ja erilaisessa työympäristössä. Siinä mielessä niiden vertailu on hyvä tilaisuus tarkastella työtapojen tehokkuutta. Toisaalta tekijöiden kokemus oli myös kasvanut toista esimerkkiä, Unelmalomaa tehdessä, joten tämä on otettava huomioon.

Opinnäytetyön päätarkoitus on tarkastella case-esimerkkeinä toimivien pelien toteutusprosessia kokonaisuutena ja perustella omat ratkaisut, sekä etsiä parempia työtapoja tutkimalla perusteellisesti pelialalla yleisiä käytäntöjä. Toisaalta toivon, että työskentelystä kertominen antaa lukijalle kuvan siitä, millaisia haasteita peligraafikko saattaa kohdata pienessä tiimissä toimiessa.

Pelit toteutettiin 2D-grafiikalla, mutta 3D-grafiikkaengineä hyödyntäen. 3D-grafiikka on tähän tarkoitukseen yllättävän tehokas tapa, mutta käyttökelpoisten tekniikoiden etsimiseen meni ainakin itselläni jonkin verran aikaa. Käyn sen takia läpi erityisesti 2D-grafiikan tekemiseen liittyviä ongelmia, kun käytetään Photoshopia, 3D Studio Maxia ja reaaliaikaista 3D-grafiikkaa. Esitellyt työtekniikat ovat sovellettavissa esimerkiksi Unity:ssä.

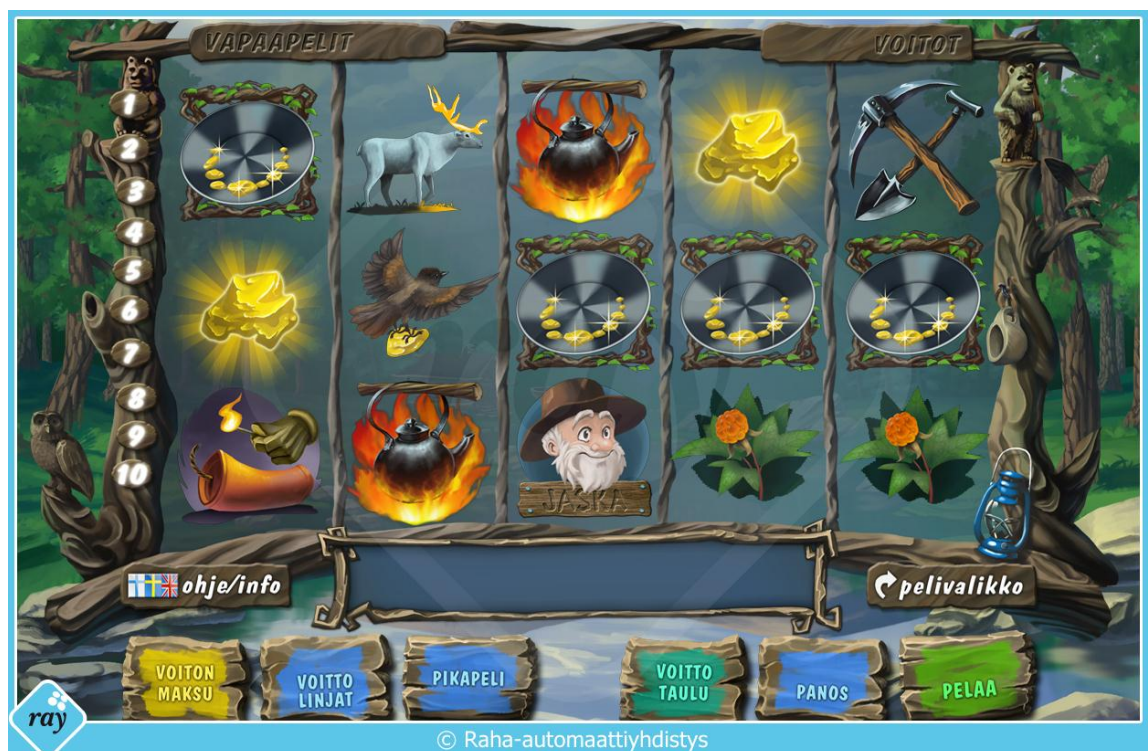
2 Huomioitavaa tekijänoikeuksista

Osa opinnäytetyössä olevasta kuvamateriaalista on erikseen merkitty Raha-automaattiyhdistyksen tunnuksilla. Tämä kuvamateriaali on tekijänoikeus- ja tavaramerkkilain alasta. Kuvamateriaalin kaikenlainen käyttö, mm. kopiointi, uudelleenkäyttö tai julkaiseminen on kielletty ilman erillistä lupaa Raha-automaattiyhdistykseltä.

3 Case-esimerkkien esittely

3.1 Kulta-Jaska

Kulta-Jaska oli RAY:n ensimmäinen selvästi monilinjaiseksi videohedelmäpeliksi erottuva peli. Sen kanssa kokeiltiin monia uusia tekniikoita ja luotiin pohjaa 3D-tekniikan hyödyntämiseen 2D-grafiikan esittämisessä. Pelin suunnittelu aloitettiin marraskuussa 2009 ja varsinainen toteutus päättyi helmikuussa 2010. Peli julkaistiin maaliskuussa 2010. Kulta-Jaska oli kaupallinen menestys ja sai muutenkin varsin myönteistä asiakaspalautetta.



Kuva 1. Mockup-ruutukaappaus Kulta-Jaskasta.

3.2 Unelmaloma

Unelmaloma oli RAY:n kolmas monilinjainen videohedelmäpeli. Kulta-Jaskan jälkeen tehdyn Potti-di-Maren, (jota en ollut mukana tekemässä) jälkeen työtavat olivat vakiintuneet ja kehittyneet, joten toteuttaminen oli kaiken kaikkiaan Kulta-Jaskaa sujuvampaa. Pelin graafisissa elementeissä näkyy lopulta monen graafikon kädenjälki, vaikka teinkin suurimman osan työstä. Unelmaloman suunnittelu aloitettiin toukokuussa 2010 ja toteutus oli valmis saman vuoden syyskuussa. Peli ilmestyi tammikuussa 2011 ja on toistaiseksi ollut kohtuullisen suosittu.



Kuva 2. Mockup-ruutukaappaus Unelmalomasta.

4 Työympäristö ja tiimityöskentely

4.1 Pelien projektihallinta

Unelmaloma tehtiin käyttäen Scrum-nimistä projektihallinnan menetelmää. Scrumia käytetään perinteisesti ketterässä ohjelmistokehityksessä. Scrum tiimi koostuu Product ownerista, Scrum-masterista ja varsinaisesta toteutustiimistä, jonka koko on 5-9 henkilöä. Toteutustiimi on itseään organisoiva, eli saa päättää keskenään millä tavalla ja missä järjestyksessä tehtävät pitäisi toteuttaa. (Schwaber & Sutherland 2010.)

Product ownerin tehtävät vastaavat suunnilleen tuottajan roolia. Hän vastaa tuotteen työlistan ylläpidosta, tehtävien tärkeysjärjestyksestä ja kaupallisesta kannattavuudesta.

Scrum master on tavallaan projektipäällikkö, mutta toteutustiimin itseään organisoivan luonteen takia, hänellä ei ole käskyvaltaa tiimiä kohtaan. Scrum master sen sijaan esittää suosituksia ja keskustelee tiimin kanssa, jos sillä on ongelmia. Toisaalta hän myös toimii yhdyssiteenä muiden tiimien, Product ownerin ja yrityksen muiden rakenteiden kanssa. Toteutustiimi vastaa tuotteen toteuttamisesta ja Scrum master siitä, että toteutuksella ei ole esteitä tai hidasteita.

Scrumin toteutus puhtaimmillaan edellyttäisi, että kaikki toteutustiimin jäsenet olisivat taidoiltaan tasavertaisia, eli pystyisivät tekemään tarvittaessa minkä tahansa tiimin työlistassa olevan tehtävän. Tässä mielessä graafikoista ja ohjelmoijista koostuva toteutustiimi ei ole aivan scrumin ideologian mukainen. Unelmaloman toteutustiimi koostui kolmesta ohjelmoijasta ja yhdestä graafikosta eli itsestäni. Tekemisen loppuvaiheilla aika meinasi loppua kesken, joten osan graafisista elementeistä toteutti muista tiimeistä apujoukoiksi tulleet kaksi ylimääräistä graafikkoa.

Scrumin perusajatuksia on toteutusprosessin läpinäkyvyys. Tämän takia siihen kuuluu jatkuva työn etenemisen tarkkailu kahden viikon välisissä sprinttipalavereissa sekä päivittäisissä toteutustiimin Scrum-tapaamisissa. Toteutustiimi antaa jokaiselle työtehtävälle aika-arvion ja niiden perusteella voidaan tarkkailla toisaalta koko projektin etenemistä, niin myös tiimin aika-arvioiden paikkansapitävyyttä. Aika-arvioiden tekeminen ja seuraaminen oli aluksi hyvin stressaavaa, mutta niistä alkoi olla iloa, kun ymmärsin,

ettei tarkoitus ole tehdä alusta asti täydellisiä arvioita ja täydellisiä arvioiden mukaisia suorituksia, vaan oppia koko ajan arvioimaan tilanteita ja tehtäviä paremmin.

Scrum tuntuu jatkuvasti tuovan omia työn suunnittelun heikkouksia esiin, enkä tunne vieläkään täysin hallitsevani asioita kyseisen projektihallinnan alla. Toisaalta olen tyytyväinen, kun ole huomannut paljonkin konkreettisia asioita, joita minun pitäisi kehittää omassa työskentelyssäni. Välillä mietin kuitenkin missä määrin aika-arvioiden tekemisestä on hyötyä esimerkiksi suunnitteluvaiheessa. Liiallinen tuntien ja päivien tuijottaminen johtaa lopulta pakotettuihin ja ehkä turhiinkin kompromissiratkaisuihin. Kyse lienee kuitenkin ennen kaikkea siitä, etten osaa vielä arvioida työtehtävien suurpiirteisyyttä. Koko peliä koskevan konseptikuvituksen ja tyylin etsiminen on kuitenkin täysin eri asia, kuin vaikkapa yksittäisen hahmon toteutus ja animointi.

Kulta-Jaska toteutettiin samantapaisella projektihallinnalla, mutta Scrum Masterin sijaan toteutustiimin yksi jäsen oli projektivastaava. Aika arvioita ei juurikaan tehty ja virallisia tapaamisia pidettiin vain kahden viikon välein. Tekeminen oli huomattavasti vapaamuotoisempaa. Vaikka muistelenkin lämmöllä sitä, että sain tehdä aina kaksi viikkoa omassa rauhassa töitä, niin oma toimintani toteutusvaiheessa oli Kulta-Jaskaa tehdessä kuitenkin huonommin suunniteltua. Varsinkin tiimikokojen kasvaessa Scrumin kaltaista tarkempaa projektihallintaa todennäköisesti tarvitaan.

4.2 Muut työskentelyyn vaikuttavat tekijät

En usko että yhtäkään peliä on koskaan luotu täydellisissä olosuhteissa. Siinä mielessä työskentelyyn vaikuttavia ulkopuolisia tekijöitä on tavallaan turhaa käydä läpi, sillä seuraavassa projektissa on taas omat yllätyksensä odottamassa. Omia aika-arvioita ja työskentelyä seuraamalla voi kuitenkin oppia jonkin verran ennakoimaan ongelmatilanteita.

Molemmissa esimerkkiprojekteissa parhaita puolia oli tasapainoinen ja stressitön työympäristö. Lähes kaikki stressiä aiheuttavat tekijät olivat lopulta omia valintoja esimerkiksi toteutustapojen suhteen.

Kulta-Jaskaa tehdessä olosuhteet olivat henkilökohtaiselta kannalta todella suotuisat, koska aloin sitä tehdessä viimein tottua uuden työpaikan käytäntöihin. Energiaa tekemiseen riitti ja pelin aihepiiri oli itselle tuttu ja mieluista. Pelin suunnitteluprosessi lähti rauhallisesti käyntiin ja ehdin tutustua referenssimateriaaliin huolella. Ainoa ongelma oli se, että peliä tehdessä joutui kokeilemaan paljon uusia tekniikoita. Toteuttamisessa ei ollut vielä rutiinia ja virheitä tuli silloin tällöin tehtyä, mutta mihinkään suuriin ongelmiin ei törmätty.

Unelmaloman suhteen tilanne oli lähes päinvastainen. Tulin suoraan koulusta tekemään pelin grafiikoita kesän ajaksi ja olin ollut poissa työpaikalta monta kuukautta. Pelin suunnittelu oli aloitettu jo aiemmin, joten jouduin tutustumaan valmiiksi suunniteltuun materiaaliin ja jatkamaan sen pohjalta. Projekti oli siis jo tavallaan hyvässä vauhdissa sillä hetkellä, kun tulin siihen mukaan. Tämä yhdistettynä siihen, että muut tiimin työntekijät viettivät aina välillä kesälomiaan aiheutti sen, etten ollut läheskään tarpeeksi selvillä projektin tarpeista ja alkuperäisistä suunnitelmista.

Projektinhallinta oli muuttunut Scrumiksi ja siihen totuttelu vei taas aikansa. Toisaalta poissa ollessani teknisiä työtapoja oli viety huomattavasti toimivampaan suuntaan, joten varsinainen toteutus oli paljon aiempaa sujuvampaa.

Suurin ongelma tekemisessä oli oma väsymys, joka johtui siitä etten ollut pitänyt varsinaista lomaa kahteen vuoteen. Tilanne oli itselleni uusi, joten en tiedostanut levon ja virkistytymisen tarvetta tai työn todellista haastavuutta. Väsymys aiheutti sen, että ajatus ei kerta kaikkiaan kulkenut aivan tavalliseen tapaan ja päätösten teko vaikeutui. Työn tekeminen muuttui hitaammaksi, minkä taas korvasin ylitöillä. Lopulta omat ratkaisut muuttuivat yhä turvallisuushakuisemmiksi ja tekeminen alkoi olla yhä pakotetumpaa. Mukavassa työympäristössä tällainenkin tilanne oli oikeastaan täysin siedettävissä ja ongelmien tullen apuja sai kuitenkin aina. Jälkikäteen katsottuna oli tietenkin vastuutonta mennä projektiin täysillä mukaan tällaisessa olotilassa, enkä enää samaa virhettä toistaisi. Tärkein opetus tässä olikin omien rajojen löytäminen ja konkreettinen näyttö siitä, miten oleellinen osa mielen pitäminen virkeänä on luovassa työssä.

5 Toteutusvaiheet

5.1 Suunnittelu

Varsinainen peli-idea on yleensä päätetty suunnitteluvaiheen ulkopuolella ja sitä ruvetaan tässä vaiheessa Product ownerin johdolla miettimään tarkemmin. Suunnitteluvaiheessa pidetään silloin tällöin palavereja ja ideariihiä, joissa kehitetään mahdollisimman paljon vaihtoehtoisia ideoita ja toisaalta sovitaan lopulliset linjat ja merkitään peliin halutut ominaisuudet tuotteen työlistaan. Muuten suuri osa suunnittelusta tapahtuu itsenäisesti mutta niin, että aiemmin yhdessä sovittuja linjauksia seurataan.

Graafikon kannalta tässä vaiheessa kannattaa kerätä referenssimateriaalia ja muutenkin tutustua pelin aihepiiriin mahdollisimman huolellisesti, jotta voisi varmasti katsoa asiaa omaperäisestä näkökulmasta.

Ennen toteutusta tehdyn suunnitelman olisi syytä olla mahdollisimman runsas. Ideoita on syytä olla liikaakin, kunhan ne voi laittaa tärkeysjärjestykseen. Kulta-Jaskassa ominaisuuksia ja ideoita päädyttiin karsimaan paljonkin alkuperäisistä suunnitelmista. Unelmaloman kohdalla taas kesken toteutusta saattoi huomata, että työlistan ominaisuuksia ei ollut määritelty tarpeeksi yksityiskohtaisesti. Toteuttamisen pysähtyminen yhtäkkiseen suunnitelmatyhjiöön tekee työskentelystä epävarmaa ja sekoittaa työtahdin koko tiimin osalta. Toteutuspainneiden ollessa päällä, suunnittelutyöstä tulee hätäistä ja ilotonta, mikä johtaa huonompiin ja jopa pelkurimaisiin ratkaisuihin.

Pelin tekeminen ei kuitenkaan kaatunut tähän, koska apujoukkoja riitti ja ylimääräiset suunnittelupalaverit järjestettiin huolella. Muutenkin ylimääräiset suunnitteluvaiheet vaikuttivat tavallista vähemmän tiimityöskentelyn tehokkuuteen, sillä osa tiimistä oli jatkuvasti poissa kesälomien takia. Ratkaisu ja lopputulos olivat silti kaukana ihanteellisesta. Tällainen virhe olisi todennäköisesti ollut kohtalokas, jos tekijöinä olisi ollut pienempi yritys ja tiimi.

Yleensä alkuvaiheessa päätetyt ominaisuudet pysyvät kyllä loppuun asti, mutta niiden toteutuksen taso päätetään tarkemmin vasta silloin, kun pelin tärkeämpiä ominaisuuksia on jo saatu toteutettua. Jokaisessa pelissä on yleensä jotain uutta tekniikkaa, joka pitää testata. Käytännössä koko pelin millintarkka suunnittelu olisi mahdotonta, koska ongelmia ja yllätyksiä tulee aina eteen ja ne syövät resursseja.

5.2 Konseptitaide

Konseptitaidetta tehdään yleensä suunnitteluvaiheen yhteydessä. Samalla kun pelin pääominaisuuksia ideoidaan paikalleen, on graafikolla aikaa tutustua pelin aihepiiriin tarkemmin. Konseptitaidetta tehdessä on mahdollisuus irtautua valmiiksi kirjatusta vaatimuksista ja kokeilla uusia lähtökohtia. Samalla voi syventää itselleen tai muulle tiimille pelin maailmaa ja tarinaa.

Konseptitaiteen luominen on usein omalla kohdallani kaikkein stressaaviin osuus. Valmiita ideoita ei ole lyöty lukkoon ja kaikki on epävarmaa. Samaan aikaan muu tiimi saattaa joutua odottamaan päätöksiä, jos pelin pohjatyö on muuten valmiina. Placeholder-grafiikkaa on kuitenkin parempi lähteä tekemään vasta huolellisen konseptikuvilla kokeilun jälkeen. Muuten koko tiimi joutuu ehkä tuhlaamaan aikaa huonon idean toteuttamiseen.

Unelmalomaa tehdessä konsepti- ja suunnitteluvaihe jäivät valitettavan vähälle huomiolle ja peliä lähdettiin käytännössä heti tekemään placeholder-grafiikalla. Vapaan ideointivaiheen selkeä erottaminen tekemisprosessista jäi puuttumaan ja lopputuloksena oli graafisesti tylsä ja kliseisiin ratkaisuihin turvautuva toteutus. Konseptivaiheen huono huomioonotto tuntui koko projektin ajan epävarmuutena pelin perustuksia kohtaan ja lisäsi työskentelyn stressaavuutta huomattavasti aina pelin valmistumiseen asti. Konseptivaihe jäi vähälle huomiolle paitsi oman amatöörimäisyyden, myös sen takia että pelien tekemisen oletettiin olevan jotenkin rutiininomaisempaa kuin aiemmin.

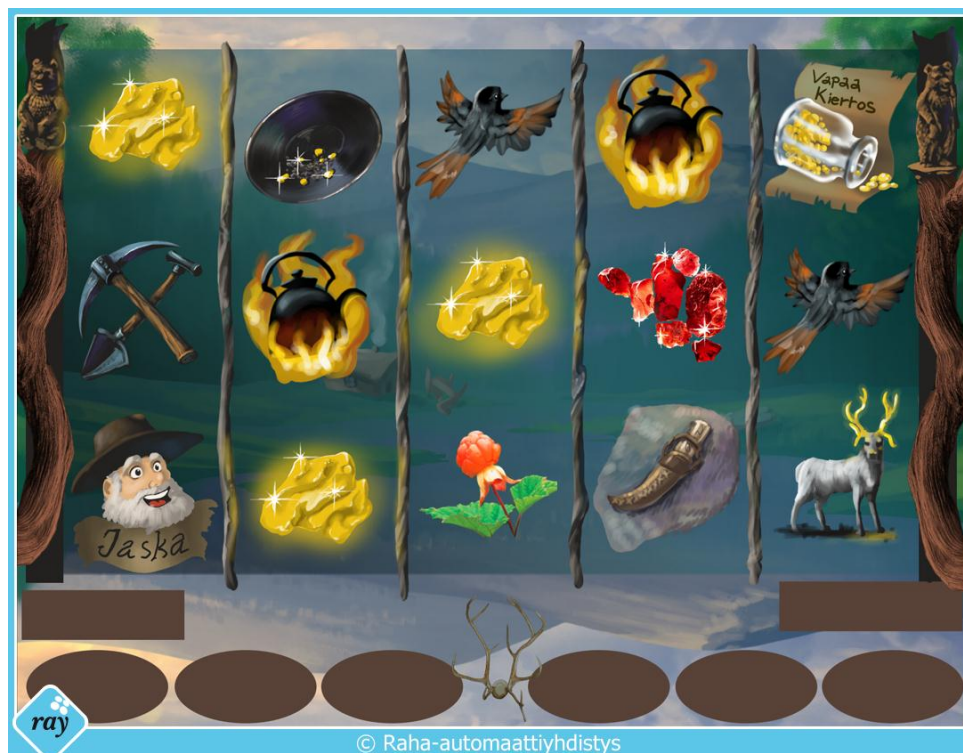
Unelmaloman grafiikan suunnittelua vaikeutti erityisesti se, että eräs toinen graafikko oli tehnyt peliin jo laadukkaita konseptikuvia. Kyseinen graafikko oli kuitenkin poistunut projektin parista ja pidin hänen tekemiään kuvia lähtökohtana suunnittelulle. Vaikka kuvat olivat laadukkaita, en ehtinyt päästä jyvälle niiden tekotavoista ja päädyin käyttämään paljon aikaa siihen, että tutkin näiden kuvien tyyliä ja sisältöratkaisuja. Uuden tyylin opettelu näin rajallisessa ajassa ja näin vähäisen materiaalin perusteella oli kuitenkin mahdottomuus, joten olisin päässyt nopeammin vauhtiin jos olisin aloittanut koko homman nollapisteestä. Kuvittelin että olisin nopeuttanut työtä käyttämällä osit-

tain jo tehtyjä kuvia, mutta käytännössä ratkaisu aiheutti päänvaivaa ja oikean tyylin etsimistä projektin loppuun asti.

5.3 Placeholder-grafiikka

Placeholder-grafiikan ensisijainen tarkoitus on tarjota ohjelmoijille pelin tarvitsemat graafiset palikat, jotta pelin tekniikan toimivuus voidaan testata. Näin saadaan varsin hyvä kuva siitä, millä tavoin peli kuluttaa laitteiston resursseja. Toisaalta koko tiimi saa myös paremman käsityksen siitä, millainen määrä graafisia elementtejä lopulta tarvitaan. Suunnitteluvaiheessa saattaa helposti jäädä huomaamatta jokin toteutusta vaativa asia.

Parhaimmillaan placeholder-grafiikka myös vastaa jollain tapaa pelin tulevaa, lopullista ulkoasua, jopa siinä määrin, että placeholder-kuvat voi hätätilanteessa jättää pelin julkaisuversioon. Placeholder-materiaalia voi tietenkin päivittää ja iteroida monta kertaa ennen lopullista versiota ja toisinaan raja näiden kahden välillä hämärtyy.



Kuva 3. Kulta-Jaskan placeholder-grafiikkana käytetty konseptikuva. Luonnosmaisuudesta huolimatta koko tiimillä oli alusta asti käsitys pelin graafisesta sisällöstä ja värimaailmasta.

5.4 Grafiikan suunnittelun periaatteita

5.4.1 Kilpailijoiden huomioonotto

Suomessa ei tietenkään voi puhua suorasta kilpailusta rahapeliä välillä, sillä Raha-automaattiyhdistys on monopoliasemassa. Nettipelien yleistyessä ihmisten tietoisuus alan yleisestä tasosta kuitenkin kasvaa koko ajan, joten selvästi ajastaan jäljessä olevat pelit menettävät varmasti huomioarvoaan. Rahapelit ovat yleensä valmistajasta riippumatta hyvin samankaltaisia keskenään, joten massasta erottumiseen kannattaa käyttää suunnittelu-aikaa. Tietenkin kilpailijoihin vertaaminen antaa aina objektiivisemmän kuvan omasta suorituksesta.

Suomessa peleissä on lainsäädännön vuoksi tavallista tiukemmat rajoitukset panosten ja maksimivoittojen suhteen. Esimerkiksi nettipeleihin verrattuna mahdollinen voitotosumma ei siis voi olla yhtä suuri houkute pelaamiseen. Tämän takia pelin muulla toteutuksella on ehkä tavallista suurempi rooli.

Kulta-Jaskaa ja Unelmalomaa tehdessä pyrin tekemään samantasoista grafiikkaa kuin internetissä pelattavissa flash-pohjaisissa monilinjaisissa peleissä. Kaikkien pelikuvioiden animointi oli oleellinen tapa erottua kilpailijoista. 3D-grafiikan hyödyntäminen antoi myös mahdollisuuksia erottua useista nettipeleistä.

Nykyisin nettipeleissäkin alkaa olla yhä enemmän animoitua materiaalia, esimerkiksi esirenderöidyn videokuvan muodossa. Osa peleistä on selvästi tehty kovemmilla resursseilla kuin Kulta-Jaska tai Unelmaloma, joten pelkkä yksittäisen graafikon taitojen viilaaminen ei ehkä kauaa riitä, vaan myös tekijöiden tiimikokoja pitäisi kasvattaa.

5.4.2 Automaattien laitteisto ja ympäristö

Raha-automaattiyhdistyksen pelit pyörivät hyvin vaihtelevissa ympäristöissä. Automaatteja on hämärissä baareissa, kirkkaasti valaistuissa marketeissa ja muissa vähemmän kuin ihanteellisissa ympäristöissä. Nykyiset videopeliautomaatit perustuvat kosketusnäyttöihin, joihin kerääntyy lopulta kerros kaikkea, mitä sormet kantavat mukanaan. Näyttöjen pinta saattaa näistä syistä heijastaa paljonkin ympäristön valoa tai suoras-

taan estää näkemistä. Lisäksi näyttöjen kalibroinnissa, iässä ja laitteistossa saattaa olla eroja.

5.4.3 Asiakaskunta

Vaikka pelit suunnataankin yleensä tietyille kohderyhmille, voidaan olettaa, että todella laaja yleisö tulee lopulta ainakin kokeilleeksi pelejä. On syytä ottaa huomioon, että pelejä saattaa pelata heikkonäköinen ihminen. Yleinen selkeys on muutenkin tärkeää, kun otetaan huomioon eri-ikäiset pelaajat, joista kaikilla ei ole kokemusta tietokoneiden tai pelien käyttöliittymistä.

5.4.4 Pelien selkeys ja elementtien tärkeysjärjestys

Aiemmin mainittujen syiden takia grafiikan selkeyteen pitäisi kiinnittää erityistä huomiota. Monilinjaisissa peleissä tärkein erottelu tapahtuu käyttöliittymän eli nappien, taustaelementtien ja varsinaisten pelikuvioiden välillä. Tämä tapahtuu luonnollisesti rakentamalla kontrasteilla, värien voimakkuudella, terävyydellä ja yksityiskohtien tiheydellä selkeä hierarkia.

Kulta-Jaskassa ja Unelmalomassa pelikuvioina käytetään poikkeuksellisesti pelkkiä kuvia, eikä pelikorteista tuttuja 10, J, Q, K ja A –merkkejä. Tyypillisesti monilinjaisissa peleissä käytetyt pelikorttimerkit helpottavat kuvioiden keskinäisen arvojärjestyksen hahmottamista. Pelkkiä kuvia käytettäessä on tärkeää kiinnittää huomiota kuvien keskinäiseen sisältöön, värikkyyteen, kokoon ja huomioarvoon yleensä.

Pelikuviot olisi hyvä kyetä tunnistamaan jo kaukaa ja pelkästään silhuetin perusteella. Yksi pelikierros kestää vain viitisen sekuntia, joten peliä on jännittävämpi seurata, jos kykenee erottamaan ruudulta sekunnin murto-osassa millaiset mahdollisuudet voittoon on. Koska kuviot ilmestyvät ruudulle satunnaisessa järjestyksessä, niiden pitäisi myös muodostaa kaikissa tapauksissa esteettisesti miellyttävä kokonaisuus yhdessä.

Kuviot kannattaa suunnitella heti alussa huolella ja katsoa niitä yhdessä siinä koossa, missä ne näkyvät pelissä. Minkään kuvion animointia tai viimeistelyä ei yleensä kannata aloittaa ennen kuin kaikki kuviot on suunniteltu ja saatu näyttämään hyvältä keske-

nään. Vanhempien automaattien näyttöjä ajatellen väreistä kannattaa tehdä hieman tavallista kirkkaammat.

Unelmalomassa graafinen tyyli poikkesi Kulta-Jaskasta paljon jo aiemmin mainittujen syiden vuoksi. Suurin syy maalauksellisen tyylin hylkäämiseen oli kuitenkin selkeyden tavoittelu. Unelmalomassa näkyy kerrallaan suurempi määrä kuvioita, joten kuvioiden koko on myös pienempi. Tämän takia oli tärkeää käyttää mahdollisimman tasaisia väripintoja ja pikselintarkkaa jälkeä. Vaihtoehtona olisi ollut kuvioiden yksityiskohtien vähentäminen, mutta tällöin kuvien sisältö olisi kärsinyt.

Unelmalomassa käytin kuvioiden keskinäiseen tasapainottamiseen huomattavasti enemmän aikaa kuin Kulta-Jaskassa. Vaikka henkilökohtaisesti pidänkin vähemmän Unelmaloman ulkoasusta, on se kuitenkin harkitumpi Kulta-Jaskan. Unelmaloman kuviot ovat myös pienemmästä koostaan huolimatta sisältörikkaampia kuin Kulta-Jaskan vastaavat. Kuvien ja pelin muodostama kokonaisuus ei vain toimi Unelmalomassa yhtä hyvin.

5.4.5 Tarinankerronta staattisessa peliympäristössä

Monilinjaisissa peleissä valtaosa pelin tapahtumista tapahtuu yhdessä, staattisessa peliruudussa. Välillä vaihtelua luodaan bonuspeleillä, mutta niihin pääsy on sattumanvaraista eikä niiden kesto ole yleensä kovin pitkä. Käytännössä voidaan olettaa että pelaaja näkee pelistä pelkän peruspelin.

Peliin on siis hyvin vaikea saada jatkuvuuden tunnetta tavallisten pelien tapaan. Tämän takia tarina on tavallaan piilotettava pelikuvioihin ja taustaelementteihin. Tarinaa ja pelin maailmaa ei voi suoraan selittää eikä lineaarinen kerronta ole mahdollista. Pelin tarina tavallaan pomppii samoissa tapahtumissa ajattomasti, ikään kuin kelaasi elokuvasta lempitapahtumiaan.

Käytännössä tarinaan voi vihjata kuvioissa näkyvillä asioilla ja niiden animaatioilla. Toisaalta taustakuvaan ja kehyselementteihin voi sijoittaa hyvinkin yksityiskohtaisia kuvituksia, jotka avaavat pelin maailmaa. Kerrontaan ja teemaan liittyvät yksityiskohdat

pitää kuitenkin jättää aina toissijaiseksi, jos ne vievät liikaa huomiota varsinaiselta peliin liittyvältä informaatiolta.

Koska peliympäristö on niin staattinen, voidaan olettaa että pelaajan silmät lopulta ekeysyvät katsomaan pieniäkin yksityiskohtia. Parhaimmillaan pelin ikä pitenee, kun pelaajalla on mahdollisuus löytää kuvituksesta aina jotain uutta. Voittoanimaatiot toimivat jo itsessään houkuttimena, sillä on hyvin epätodennäköistä että näkisi kaikkia animaatioita yhden pelikerran aikana.

5.4.6 Ajoitus, rytmitys ja huomion kohdistaminen

Pelaajan huomion kiinnittäminen pelin kannalta oleelliseen informaatioon tapahtuu väistämättä myös ajoituksen kautta. Muutaman sekunnin pelikierroksen aikana voi tapahtua paljonkin asioita, jotka pitää saada selkeästi näkyviin. Joskus erityisiä voittomahdollisuuksia, kuten bonuspeliin pääsyn mahdollisuutta hehkutetaan jo kesken pelikierroksen.

Pelikierroksen jälkeen tapahtuu vielä voittojen esittely. Samalla kun voittoanimaatioita pyöritetään, ruudulla joutuu ehkä kertomaan voitetuista vapaapeleistä muiden linjakohdistusten voittojen lisäksi.

Pelaajalle esitellään usein paljon informaatiota kerralla mutta kaikkien esittäminen selkeästi ajan kanssa tekisi pelin kulusta tökkivää ja hidasta. Usein voitot ovat pieniä ja esitelty informaatio pelaajan kannalta niin pienessä roolissa, että niille ei kannata uhraa kovin paljon aikaa tai odottaa erikseen pelaajalta käskyä jatkaa peliä.

Varsinaisten pelikuvien voittoanimaatioissa voi myös miettiä millä tavalla animaation pituus vastaa kuvion arvoa. Jos pelikierroksella saa monta erilaista voittoa, animaatiot väistämättä vievät tilaa toisiltaan, kun pelaaja joutuu katsomaan ne kaikki vuorotellen läpi. Tällöin usein nähdyt pienet voittoanimaatiot saattavat pitkästytää ja syödä osan animaatioiden palkitsevuudesta.

Kulta-Jaskassa ja Unelmalomassa mietittiin kyllä ajoituksia, muttei juurikaan rytmitystä. Rakentamalla voittoanimaatioihin ja informaatioteksteihin tietty rytmi, voitaisiin ehkä

tehdä pelin tapahtumien seuraamien pelaajalle paremmin ennakoitavaksi. Tällöin myös pelitapahtumien synkronointi musiikin kanssa olisi helpompaa ja pelielämys ehkä kokonaisemman tuntuinen.

5.4.7 Graafisten elementtien tärkeysjärjestys toteutuksen kannalta

Työskentelyn kannalta on oleellista olla tietoinen koko pelin halutusta ominaisuuslistasta ja ominaisuuksien tärkeydestä pelin kannalta. Peruspelin yleinen ulkoasu on tietysti tärkeintä saada ensin miellyttäväksi. Riippuen erilaisten bonuspelien esiintymistiheydestä, voidaan niiden ulkoasuun panostaa tämän jälkeen. Jos bonuspelit ovat vähäisessä roolissa, kannattaa mielummin tehdä peruspelistä mahdollisimman hiottu ja tehdä bonuspelit sillä ajalla, mikä sattuu jäämään jäljelle.

Toistaiseksi kymmenen pelikuvion muuttaminen stillikuvista animoiduiksi kuvioiksi on vienyt noin kuukauden koko tekoajasta. Animoinnin suunnittelu ja materiaalin valmistelu sitä varten vie niin paljon aikaa, että on syytä harkita missä vaiheessa rupeaa työstämään tätä osuutta, jos sille jää aikaa ollenkaan. Kaunis maalattu staattinen kuva voi hyvinkin olla tehokkaampi kuin heikkotasoinen animoitu kokonaisuus.

Pelien käyttöliittymä toistuu usein pelistä toiseen toiminnaltaan samanlaisena, mutta hieman varioituna, joten siihen kannattaa panostaa vasta toteutuksen loppuvaiheessa. Joskus toisesta pelistä voidaan kierrättää onnistuneesti käyttöliittymäelementtejä ja saadaan näin käytettyä energiaa pelin muun ulkoasun hiomiseen.

Logo kannattaa samaten tehdä vasta viimeisten asioiden joukossa. Logo toimii käytännössä samalla pelin kansikuvana ja siihen saa parhaiten kiteytettyä toteutuksen aikana syntyneet ideat, kun suurin osa pelistä on jo valmiina. Logoa saattaa olla myös mahdollisuus vielä viilata, kun peli on testauksessa ja varsinainen toteutusvaihe päättynyt.

5.4.8 Käyttökelpoisuus pelin markkinointiin

Peligraafikon tehtävä ei ole miettiä pelin markkinointia, eikä siihen yleensä varata juurikaan erikseen aikaa. Markkinointiosasto vastaa logoa lukuun ottamatta markkinointiin liittyvän materiaalin tuottamisesta. Siitä huolimatta pelin ja markkinoinnin muodostama

kokonaisuus on todennäköisesti yhtenäisempi, jos varsinaista peligrafiikkaa on helppo käyttää markkinointiin. Käytännössä peligraafikon on siis hyvä miettiä, missä määrin pystyy valmistelevaan grafiikkaa erikokoisia painatuksia tai videokuvaa varten.

6 Grafiikan tekninen toteutus

6.1 Peligraafikon suhde tekniikkaan ja peliohjelmaan

Peligrafiikkaa tehdessä optimointi on erityisen tärkeää ja graafikon on tärkeää olla tietoinen siitä millaisella tekniikalla lopullinen peli pyörii. Jos tekniikkaa ei onnistu hyödyntämään optimaalisesti tuloksena on joko huonosti pyörivä tai huonommalta näyttävä peli. (Ahearn 2009.) Raha-automaattiyhdistyksen pelejä tehdessä joutui totuttelemaan tiettyihin kehitystyökaluihin ja työtapoihin, mutta pelialalla ei ole tavatonta, että jo yksittäisellä pelillä olisi täysin omat kehitystyökalunsa. 3D-työkalujen käyttäminen 2D-grafiikan tekemiseen on suhteellisen yleistä ja siinä on jopa hyvät puolensa.

Pienessä tiimissä toimiessa kehitystyökaluihin ja tekniikkaan tutustumiseen joutuu väistämättä käyttämään paljon omaa aikaa. Suuri osa esimerkkipelien toteutustavoista oli jollain tavalla puutteellisia oman kokemattomuuteni vuoksi. Käyn siksi läpi parannusehdotuksia käytettyihin tekniikoihin niiden esittelyn lomassa.

6.2 Aihepiirin termejä

Layer

Kuvataso Photoshopissa.

Kanava, Channel

Kuvan värikanava tai Alpha-kanava. Photoshopissa yksi layeri sisältää jo itsessään sekä värikanavat, että läpinäkyvyysinformaation. Ohjelmassa voidaan kuitenkin määrittää vielä erikseen kuvaan ylimääräisiä kanavia, jotka voidaan tallentaa esimerkiksi TGA-formaattiin erikseen.

RGB

Kuvan värikanavat, eli punainen, vihreä ja sininen väri.

Alpha

Kuvan pikselien läpikuultavuutta ilmaiseva kanava.

MIP map

Lyhenne sanoista *multum in parvo*, eli ”monta asiaa pienessä tilassa”. MIP mappaus tarkoittaa sitä, että tekstuurista luodaan monta pienempää versiota alkuperäisen tekstuurin lisäksi. Pienempiä versioita käytetään sen mukaan, minkä kokoinen täytettävän alueen pinta-ala on. Pienemmän tekstuurin käyttö tällaisessa tilanteessa on tehokkaampaa. (Ahearn 2009.)

Yleensä lopputulos myös näyttää paremmalta, koska suurikokoisen tekstuurin skaalaus pieneksi lennossa tuottaa yleensä virhekuvioita ja välkkymistä.

PNG

Lyhenne sanoista *Portable Network Graphics* (epävirallisesti *PNG's Not Gif*). PNG pakkaa kuvat tehokkaalla, häviöttömällä pakkauksella ja mahdollistaa alpha-kanavan tallentamisen. Formaatti on patenteista vapaa. (Roelofs 2011.)

TGA, Targa

Lyhenne sanoista *Truevision Advanced Raster Graphics Adapter*. TGA mahdollistaa yksinkertaisen, häviöttömän pakkauksen käytön ja alpha-kanavan tallentamisen. Formaatti on suosittu, sillä se on helppokäyttöinen, eivätkä sitä sido hankaloittavat patentit. (Wikipedia 2011.)

DDS, Directdraw Surface

Microsoftin kehittämä tiedostoformaatti reaaliaikasovellusten tekstuurien tallentamista varten. Tiedostomuoto on erittäin yleinen pelialalla. DDS-tiedostoon on mahdollista tallentaa etukäteen tekstuurin erikokoiset mip-mapit. Tiedostot voi pakata häviöllisesti käyttämällä DXT-pakkausta. Photoshopista kuvat voi tallentaa DDS-formaattiin esimerkiksi Nvidian DDS-pluginilla (Ahearn 2009.)

Formaatti tarjoaa myös erittäin monipuoliset vaihtoehdot kuvien tallentamiseen eri värisyvyyksillä.

DXT, DirectX Texture Compression

DXT on häviöllinen kuvien pakkaustekniikka, joka mahdollistaa poikkeuksellisesti kuvien pitämisen pakatussa muodossa näytönohjaimen muistissa. Parhaimmillaan kuvat voidaan pakata kahdeksasosaan alkuperäisestä koosta. Alpha-kanava on mahdollista tallentaa joko neljä- tai yksibittisenä. (Ahearn 2009.)

DXT-pakkaus tallentaa kuvan 4x4 pikselin paloissa. Jokaista palaa kohti tallennetaan vain kaksi väriä, jotka tallennetaan 16-bittisenä. Tämän takia pakkaus ei sovellu hyvin hienovaraisiin värivaihteluihin, väriliukuihin tai terävyyttä vaativaan grafiikkaan. (Linde 2005.)

Z-Buffer

Syvyyspuskuri, johon näytönohjain tallentaa jokaisen pikselin syvyysarvon kamerasta katsottuna. Syvyyspuskuriin jää vain kaikkein etummaisena näkyvä pikseli, joten näkymättömiin jääviä pikseleitä ei tarvitse turhaan piirtää. Materiaalikohtaisesti voidaan määrätä kirjoittaako materiaali Z-Bufferiin tietoa tai tarkistaako se piirrettäessä Z-Bufferia.

Alpha-blend

Piirrettävän pikselin sekoittaminen sen taustalla olevaan väriin alpha-arvon perusteella.

Alpha-test

Pikseli piirretään tai jätetään piirtämättä materiaalissa määritellyn alpha-arvon raja-arvon perusteella. Yleensä alpha ilmaistaan 0-255 arvoilla. Alpha-testaus voisi esimerkiksi jättää piirtämättä kaikki pikselit, joiden alpha on alle 128. Kaikki raja-arvon ylittävät pikselit piirretään täysin peittävinä, ilman alpha-blendausta.

Ortografinen projektio

Projektio, jossa kohteet eivät pienene horisonttia kohti, vaan viivat pysyvät yhdensuuntaisena. Projektio mahdollistaa 3D-grafiikan esittämisen kaksiulotteisen näköisenä.

Riggaus

Kolmiulotteisen objektin kontrollobjektien (luiden) tekeminen animointia tai asentojen muuntelua varten.

Skinnaus

Riggausvaiheessa tehtyjen luiden vaikutusalueiden määrittely objektissa.

Varjostinohjelma, Shader

Varjostinohjelma on pienempi ohjelma, joka käsittelee ruudulle piirrettävää grafiikkaa. Varjostinohjelmat jaetaan *verteksivarjostimiin*, jotka käsittelevät geometriaa ja verteksien ominaisuuksia ja *pikselivarjostimiin*, jotka käsittelevät yksittäisiä pikseleitä. Varjostinohjelmilla voi vaikuttaa piirrettävän grafiikan lopputulokseen käytännössä rajattomasti. (Ahearn 2009.)

Verteksivarjostimia voi käyttää esimerkiksi tuulen vaikutusten simuloimiseen, liikuttamalla puun lehtien verteksejä jonkin kaavan mukaan. Myös valaistuksen voi laskea yksinkertaisimmillaan vain verteksikohtaisesti. Pikselivarjostimet mahdollistavat esimerkiksi normal-mappien käytön ja hienovaraisemman valaistuksen.

Ruudunpäivitysnopeus, frame rate

Liikkuvan kuvan yksittäisten kuvien määrä tietyssä ajanjaksossa. Peleissä pyritään yleisesti ylläpitämään ruudunpäivitysnopeutta, joka on 60 ruutua sekunnissa. Nopeus on sama kuin useiden näyttölaitteiden virkistystaajuus.

6.3 Kaksiulotteisuuden hyvät puolet

Pienellä kehitystiimillä ja rajallisella ajalla kaksiulotteinen, piirretty grafiikka on tehokas ratkaisu persoonalliseen tarinan kertomiseen ja yksityiskohtaisen grafiikan luomiseen. 2D-grafiikka saattaa olla useille ihmisille myös helpommin lähestyttävää varsinkin verrattuna kolkkaan ja ilmeettömään pienen budjetin 3D-toteutukseen. 2D-grafiikka vie myös vähän konetehoja, mikä on hyödyllistä kun ottaa huomioon että kentällä saattaa olla käytössä useita eritasoisia ja ikäisiä laitekoonpanoja.

2D-grafiikan tekeminen yleensä on nykyään erittäin palkitsevaa sillä kyseinen tekniikka elää tavallaan uutta nousukautta. Varsinkin Flash-tyylisellä paperileike-animaatiolla on viime vuosina tehty uskomattoman näköisiä pelejä. Esimerkiksi japanilainen Vanillaware on tunnettu nimenomaan tämän tekniikan mestarillisesta käyttämisestä, viimeisimpänä tuotoksenaan mainio Muramasa: The Demon Blade (Kuva 4). Vanillawarella on omat erityisesti tämäntyyppisen grafiikan tekemiseen tarkoitettut kehitystyökalut. Heidän peleissään esimerkiksi yksi hahmo saattaa koostua yli sadasta erillisestä maalatus- ta osasta. (The Spriters Resource 2009).

Varsinaisista rahapelien valmistajista ainakin Aruze hyödyntää peleissään samantapaisia 2D-animaatioita.



Kuva 4. Muramasa: The Demon Blade on oiva todiste 2D-grafiikan elinvoimaisuudesta.

6.4 3D-grafiikan käytön perustelut

Pelit toteutettiin lopulta 3D Studio Maxia ja 3D-grafiikkamoottoria käyttäen. Pääasiallinen syy 3D-grafiikan käyttöön oli se, että tähän tarvittava tekniikka ja ohjelmisto olivat

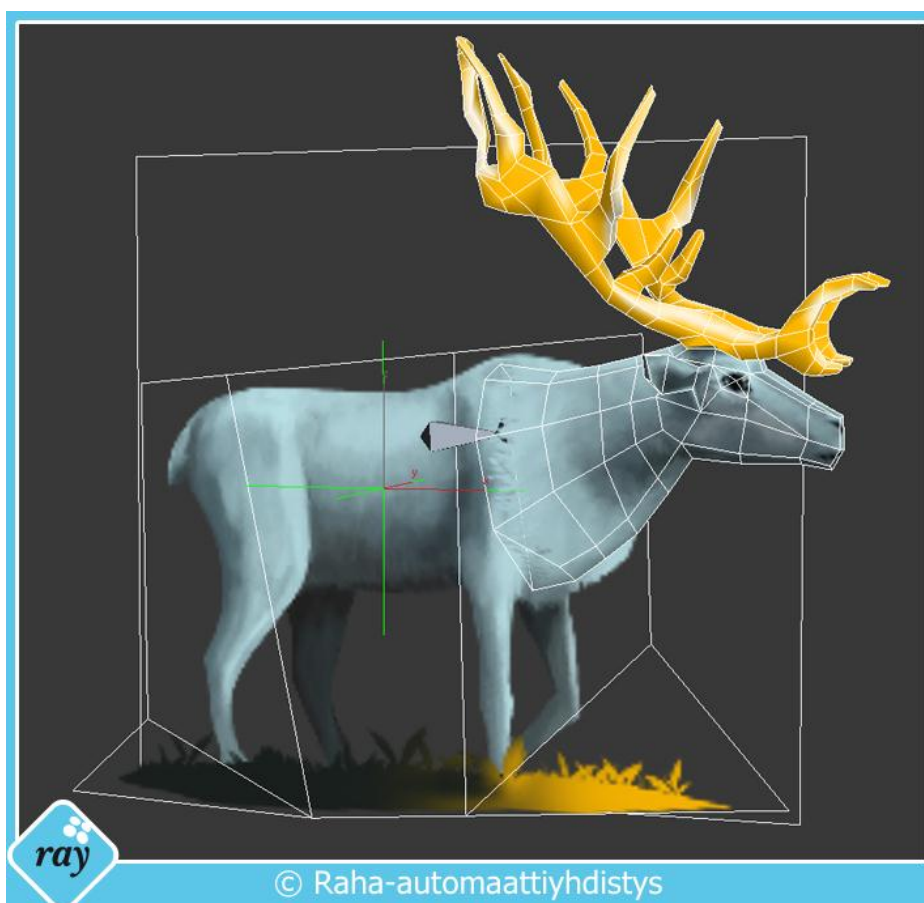
valmiiksi tarjolla. Suurin osa aiemmista peleistä oli tehty samalla tekniikalla, joten oli luonnollista jatkaa tekemistä samalla pohjalla. Toisaalta aikaa tai resursseja täysin uusien tekniikoiden kokeiluun ei sillä hetkellä ollut tarjolla.

3D-työkalujen käyttö 2D-pelejä tehdessä on sinänsä valinta tehdä asiat vaikeammalla tavalla. Esimerkiksi Flash olisi monelle taiteilijalle huomattavasti tutumpi ja kätevämpi työkalu animoidun 2D-grafiikan tuottamiseen. Toisaalta käytännössä kaikki näytönohjaimet on nykyään tehty nimenomaan 3D-grafiikkaa ajatellen, eli pelkkää Flashiä käyttämällä konetehtojen hyödyntäminen jäisi puolitiehen. Ongelmana on ennen kaikkea kehitystyökalujen puute. 3D Studio Max on jokseenkin kankea työkalu 2D-grafiikan tekemiseen, mutta sen käyttö tarjoaa toisaalta ainutlaatuisia mahdollisuuksia ja grafiikkaan saa suhteellisen vapaasti lisättyä mielenkiintoisia ominaisuuksia.

Flashiin on tosin piakkoin tulossa vaikuttavat 3D-ominaisuudet, mutta sen tehokkuus jää varmasti silti jälkeen monimutkaisen grafiikan pyörittämisessä verrattuna suoraan 3D-näytönohjainta hyväksikäyttävään grafiikkaan. Esimerkiksi Super Meat Boy -pelin animaatiot kyllä toteutettiin Flashillä, mutta vektorigrafiikka konvertoitiin peliä varten tehokkaasti pakatuiksi tekstuurilevyiksi ja animaatiot muutettiin pelimoottorin ymmärtämään muotoon. (Refenes 2009).

Sen lisäksi, että peleihin voidaan rakentaa selkeästi erottuvia kolmiulotteisia alueita vaikkapa bonuspelien muodossa, voidaan myös kaksiulotteiseksi tarkoitettua grafiikkaa käsitellä 3D-työkaluilla monipuolisemmin kuin esimerkiksi Flashin avulla. Yksi 3D:n vahvimpia puolia on vapaus skinnata animoituja objekteja oman mielen mukaan. Näin voidaan saavuttaa samanlainen vapaus 2D-grafiikan animoinnissa, kuin minkä Photoshopin ja After Effectsin CS5-versioiden mukanaan tuoma Puppet Warp -työkalu mahdollistaa. Mahdollisuus deformoida 2D-grafiikkaa vapaasti tekee animaatioista elävämmän näköisiä. 3D-grafiikan käyttö mahdollistaa myös valtavan kokoisten animoitujen hahmojen tekemisen niin, että kustannukset muistille pysyvät kuitenkin kohtuullisina.

Lisäksi 2D-animaatioihin voi yhdistää kolmiulotteisia elementtejä, mikä voi välillä lisätä uskottavuutta. Myös erilaisten näytönohjaimen varjostinohjelmien ja yksinkertaisten tekstuuriefektien käyttö lisää grafiikan mahdollisuuksia. Monimutkaisten maskattujen animaatioiden tekeminen on kuitenkin huomattavan hankalaa.



Kuva 5. Kulta-Jaskan porokuvio on staattinen 2D-kuva päätä lukuun ottamatta. Kolmiulotteinen pää oli nopea mallintaa ja mahdollisti sulavan liikkeen pään kääntyessä.

Toinen vaihtoehto olisi tallentaa animaatiot erillisestä ohjelmasta ruutu ruudulta erillisiksi kuviksi. Tällöin jouduttaisiin tallentamaan valtava määrä yksittäisiä kuvia ja vaikka ne pakattaisiin yhteen suuriksi tekstuureiksi, olisi vaikutus ladattavan kuvamateriaalin määrään ja pelin latausaikoihin vähemmän optimaalinen. Näin tallennetussa animaatioissa jouduttaisiin silti tekemään kompromisseja animaatioiden sulavuuden suhteen.

Tilaa voitaisiin toisaalta säästää tekemällä animaatioista pakattuja videotiedostoja, joita ladattaisiin aina tarpeen vaatiessa. Purkitettujen animaatioiden ongelma on niiden vaikea muokattavuus ja ajon aikana tehtävän varioinnin hankaluus. Esimerkiksi animaation hidastaminen tai kahden animaation sekoittaminen keskenään olisi mahdotonta. Tämän tyyppisiin peleihin purkitetut animaatiot voisivat kuitenkin olla täysin toimiva ratkaisu, vaikka ne olisikin ehkä haastavaa saada muistinkäytön kannalta yhtä tehokkaiksi kuin yksittäisen 3d-objektin ja tekstuurin yhdistelmä.

6.5 Läpinäkyvyydet reaaliaikaisessa 3D-grafiikassa

Useimmat 3D-grafiikkamoottorit järjestävät objektien välisen piirtojärjestyksen karkeasti niiden etäisyyden mukaan katsojasta. Virheet ovat tyypillisiä tälle piirtojärjestykselle. Esimerkiksi kamerakulman vaihdos saattaa asettaa objektit viistottain toistensa päälle, siten, että piirtojärjestystä on mahdotonta tehdä oikein. Valitettavasti myös objektien sisäinen polygonien piirtojärjestys on usein täysi mysteeri. Piirtojärjestys on erittäin tärkeä asia, kun käsitellään polygoneja, joissa on läpinäkyvyyksiä. Ilmeisenä esimerkkinä juuri piirretyn 2D-grafiikan reunat.

Näytönohjain tallentaa jokaisen pikselin syvyysarvon Z-bufferiin, eikä piirrä pikseleitä, jotka jäisivät olemassa olevan pikselin alle. Z-buffer siis tavallaan korjaa yksinkertaisemman piirtojärjestyksen jättämät virheet. Läpinäkyvyyksien kannalta Z-buffer kuitenkin hankaloittaa asioita. Jos ruudulle siis piirretään heti alkuun etualalle kuuluva puoliksi läpinäkyvä pikseli, ei sen alle enää voida piirtää asioita, joiden kuuluisi näkyä sen läpi.

Helpoin ja yleisimmin käytetty tapa kiertää ongelmaa on käyttää alpha-testausta eli alpha-kanavasta jotain raja-arvoa, minkä jälkeen pikseleitä ei enää piirretä ollenkaan. Näin Z-Bufferiin ei enää jää osittain läpinäkyviä pikseleitä haittaamaan piirtoa. Haittapuolena osittaisia läpinäkyvyyksiä on tällöin mahdotonta toteuttaa ja Alpha-kanavan reunalla oleville alueille tulee sahalaitaiset reunat (kuva 6).

Jos läpinäkyvät alueet ovat selkeästi irrallisia kokonaisuuksia, kuten ikkunalaseja tai hehkuvia erikoisefektejä ne voidaan piirtää lopulta yhdessä, kun kiinteät pinnat on jo piirretty. Tällöin läpinäkyvät alueet piirretään ilman Z-Bufferiin piirtämistä, eli niiden keskinäisellä piirtojärjestyksellä ei ole merkitystä. Läpinäkyvyyksien on kuitenkin syytä olla toiminnaltaan yksinkertaisia, esimerkiksi selkeästi kirkastavia tai tummentavia, ettei piirtojärjestyksen puute ala jälleen häiritä lopputulosta.

Kulta-Jaskan ja Unelmaloman tekemisen jälkeen on löytynyt joitain uusia konsteja piirtojärjestyksen kanssa elämiseen esimerkiksi varjostinohjelmien ja parempien mallinuskäytäntöjen avulla. Valitettavasti näiden pelien animaatiot toistuvat kuitenkin vielä rosoireunaisina.



Kuva 6. Kulta-Jaskan Emma-voittokuviossa osa polygoneista peittää kokonaan alleen jotkin asiat väärän piirtojärjestyksen takia. Oikean alakulman kuvassa näkyy, että alpha-testausta käyttämällä ongelma korjaantuu, mutta kuvaan tulee rosoiset reunat.

6.6 2D-hahmon tekeminen Photoshopissa ja 3D Studio Maxissa

6.6.1 Hahmojen maalaus ja tekstuurien kokoaminen

Kun hahmon luonnosversio on valmis, on se yleensä helppo hajottaa karkeasti layereihin ja maalata sen jälkeen puhtaaksi. Hahmon eri palaset on syytä asetella mahdollisimman tiiviiksi "palapeliksi", jotta tekstuurin pinta-ala ja muistinkäyttö pysyy mahdollisimman tehokkaana (Kuva 7). Jos tekstuuri sisältää tekstuurivaihdoksilla animoituvia elementtejä, kuten kasvonilmeitä tai muuta poikkeuksellista, kannattaa myös miettiä

miten eri palasten asettelu helpottaisi työn tekemistä 3D-ohjelmassa. Usein helposti sujuva työskentely kannattaa asettaa liiallisen teknisen optimoinnin edelle.

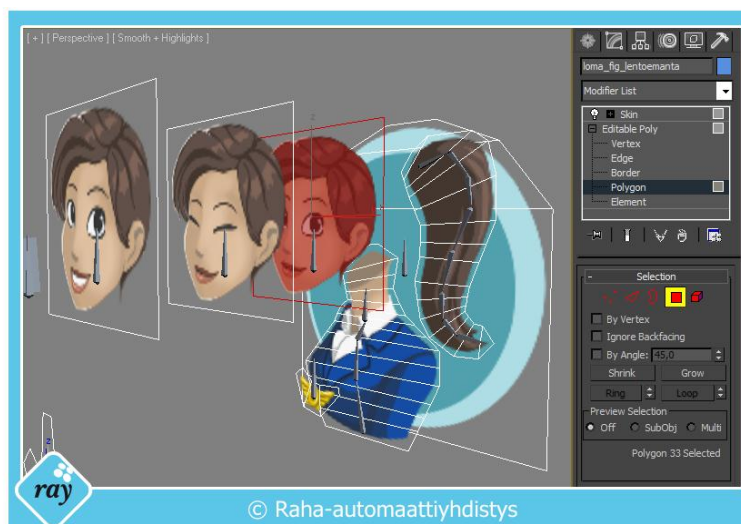


Kuva 7. Unelmaloman lentoemäntä-voittokuvion 3D-mallin tekstuuri.

6.6.2 Mallinnus, riggaus ja animointi

Näytönohjaimelle piirrettävät objektit lähetetään piirtokutsuina eli batcheinä. Piirtokutsujen lähettäminen kuluttaa prosessorin tehoa. Niiden määrä on syytä pitää alhaisena, jotta näytönohjainta voitaisiin hyödyntää mahdollisimman tehokkaasti. (Wloka 2005). Tämän takia yksi animoitu objekti on syytä pitää nimenomaan yhtenä skinnattuna objektina, sen sijaan että sitä käsiteltäisiin erillisinä palasina. Kun objekteja on vähän, on myös tekijöille helpompaa käsitellä pelin eri resursseja.

Työ alkaa käytännössä sillä, että hahmon tekstuuri asetetaan plane-objektiin, josta sen jälkeen leikataan erilleen hahmon osat. Osat asetellaan päällekkäin oikeaan järjestykseen ja yhdistetään sitten yhdeksi objektiksi *Edit Poly*-modifierin *Attach*-komennolla. Tässä vaiheessa on tilaisuus vaikuttaa objektin sisäiseen piirtojärjestykseen. Kun eri objektit yhdistää yksitellen, voi yhdistämisen tehdä siinä järjestyksessä, minkä haluaa hahmon osien lopullisen piirtojärjestyksen olevan pelissä (Kuva 8). Tämä tietysti sillä oletuksella, että pelin grafiikkamoottorin tiedostoformaatti tallentaa polygonit samassa järjestyksessä kuin 3D Studio Max. Jos hahmon animaatioissa tapahtuu palasten siirtymiä toistensa eteen tai taakse, piirtojärjestys näkyy tietenkin taas väärin.



Kuva 8. 3D Studio Maxin *Edit Poly*-modifier näyttää valitun polygonin järjestysnumeron oikeassa alakulmassa.

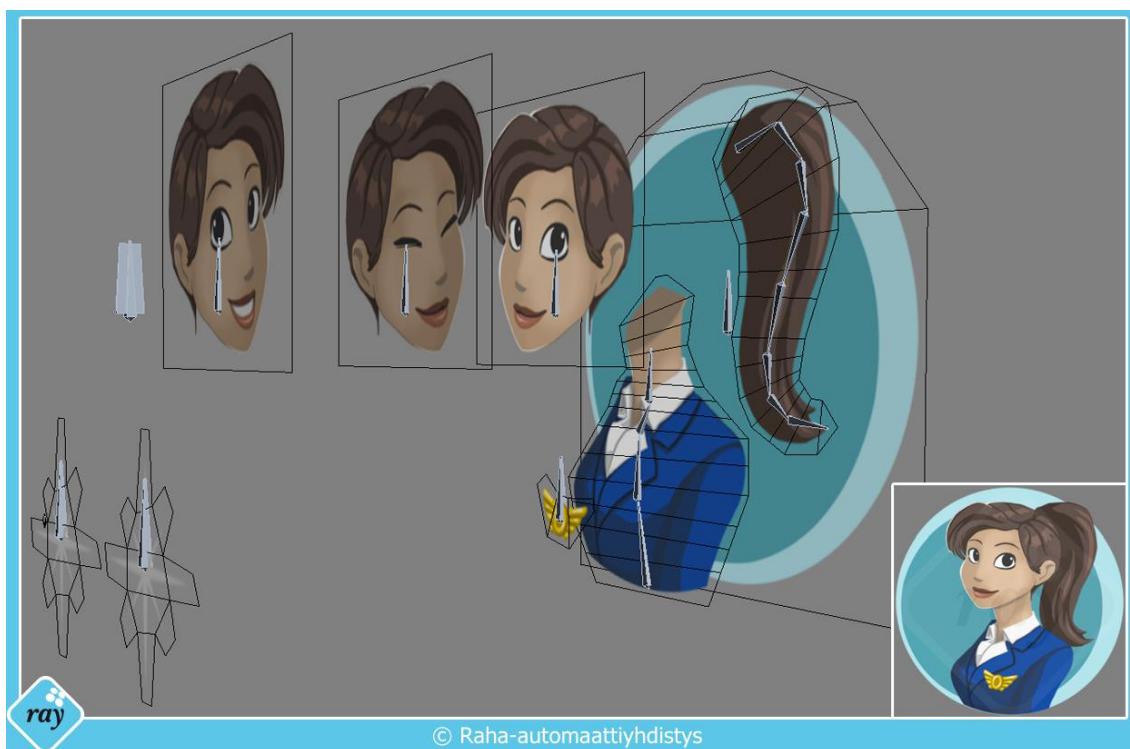
3D Studio Maxin *Edit Poly*- ja *Skin*-modifierien yhdistelmä on yllättävän joustava käyttää. Valmiiksi skinnattuun objektiin voi helposti lisätä uutta geometriaa tai muokata jo skinattua geometriaa. Objektien muokkaus jälkikäteen on siis parhaimmillaan hyvinkin helppoa. Geometriaa suuresti muokatessa ainoa ongelma on jälleen piirtojärjestyksen hajoaminen.

Pidän usein 3D-mallin eri palaset etäällä toisistaan, jotta niiden skinnaus ja animointi olisi helpompaa. Palojen etäisyyksillä ei sinänsä ole väliä, koska peligrafiikka piirretään ortografista projektiota hyödyntäen. Ongelmia voisi tulla, jos pelissä olisi monta animoitua hahmoa, jotka saattaisivat mennä päällekkäin. Kulta-Jaskassa ja Unelmalomassa tällaisia tilanteita kuitenkin ole.

Pelien animaatiot ovat hyvin lyhyitä ja yksinkertaisia. Animaatioita on usein vain yksi kappale. Tämän takia riggaukseen ei kannata käyttää paljoa aikaa. Luiden nimeämisesäkin riittää yleensä, että pelin ja hahmon nimi ilmenee ja numerotunniste erottaa yksittäiset luut toisistaan. Luita ei tietenkään näissä peleissä käytetä muuhun, kuin yksittäisen animaation tekemiseen. Koodissa ei esimerkiksi tarvita yksittäiseen luuhun liittyvää informaatiota. Tällaisissa tapauksissa nimeämisen tulisi taas olla huolellisempaa.

Tekstuurivaihdoksia vaativat animaatiot, kuten kasvonilmeet saadaan aikaan, kun tehdään tekstuuriin vaihtoehdot eri animaatiovaiheille. 3D-objektiin asetellaan nämä eri

tekstuurinpalaset tarkasti päällekkäin, ettei animaatiossa ilmenisi nykimistä. Jokaiselle palalle tehdään rigiin oma luu. Tekemällä objektin materiaalista yksipuolinen ja kääntämällä tietyt luut poispäin kamerasta saadaan piilotettua ne vaiheet, joiden ei ole tarkoitus näkyä sillä hetkellä (Kuva 9). Tekstuurivaihdosten animaation on syytä olla mahdollisimman nopea, muuten tekstuuripalaa piiloon pyöräyttävä animaatio näkyy pelissä häiritseväenä välähdyksenä. Kulta-Jaskan ja Unelmaloman animaatiot on tehty 60 ruutua sekunnissa nopeudella. Animaationopeus paitsi vastaa nopeutta, jota pelin grafiikkamoottori pyrkii ylläpitämään, myös pitää tekstuurivaihdosten yhden ruudun pituisen animaation erittäin lyhyenä. Tekstuurivaihdoksien animaatioissa olisi kätevää käyttää stepped-tangentteja, mutta pelien käyttämä grafiikkamoottori ei satu niitä tukemaan.



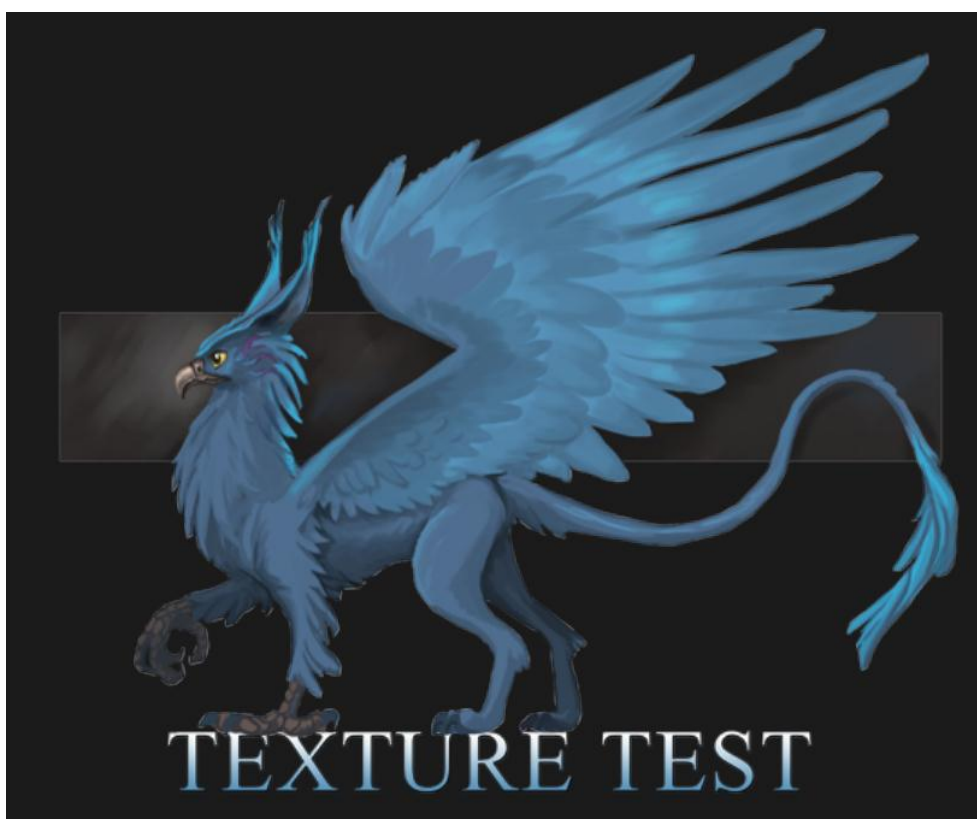
Kuva 9. Unelmaloman lentoemäntä-voittokuvion 3D-malli ja rigi. Kuvassa tummempina näkyvät osat ovat pelissä näkymättömiä, eli ne on käännetty poispäin katsojasta.

6.6.3 Photoshop ja alpha-kanavan tallennus

Photoshopissa PNG on kätevä formaatti siinä mielessä, että siihen voi tallentaa alpha-kanavan ilman sen erityistä määrittelyä. Muissa formaateissa alpha-kanavan joutuu määrittelemään channels-listaan, esimerkiksi *save selection* -toimintoa käyttämällä.

Useimmiten PNG-kuvat tallentuvat moitteettomasti, mutta välillä, varsinkin maalaamalla tehdyissä kuvissa, kuvien alpha-kanavan reunoille ilmestyy valkoista harsoa. Näin tallennettujen PNG-tiedostojen RGB-kanavia tarkastellessa huomaa, että alpha-kanavan ulkopuolelle jäävät osat täytetään usein automaattisesti valkoisella värillä. Tämä valkoinen väri taas saattaa vuotaa alpha-kanavan näkyvään osaan, silloin kun kuvaa katsotaan skaalattuna tai toisessa sovelluksessa (Kuva 10).

Toisaalta joissain tapauksissa on tarpeellista tallentaa kuvan alpha-kanavan näkyvän alueen ulkopuolelle värikanaviin jotain hyödyllistä tietoa. Tämä on tietysti mahdotonta, jos tallentaa Photoshopista suoraan PNG-formaattiin.



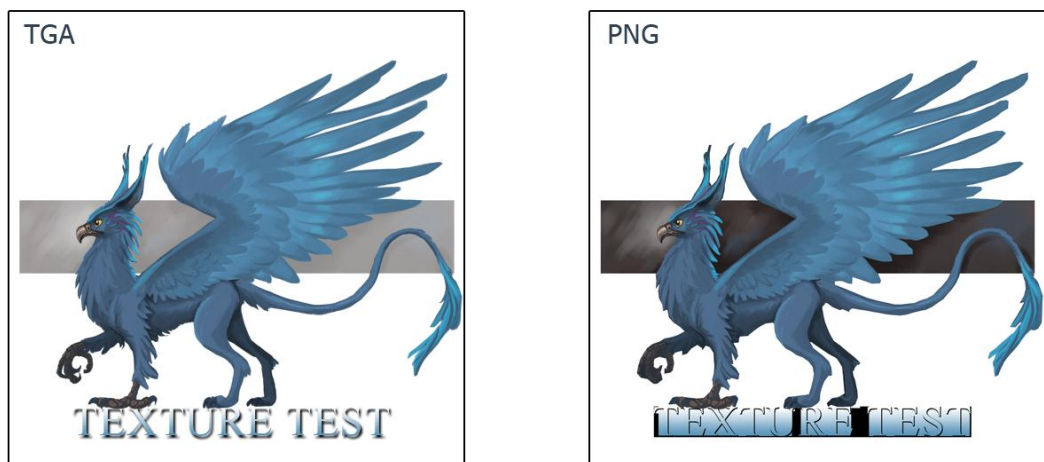
Kuva 10. Photoshopista suoraan PNG-muotoon tallennettu tekstuuri 3D-ohjelman näkyvässä. Huomaa kuvan elementtejä ympäröivä ohut, vaalea harso.

PNG:n ongelmat voi kiertää tallentamalla sen sijaan esimerkiksi TGA-formaattiin, erikseen määritellyn alpha-kanavan kanssa. Tällöin tallentaessa pitää ottaa huomioon, että kaikki Photoshopissa osittain läpikuultavat pikselit sekoitetaan valkoiseen väriin. Lopputuloksena kaikki läpikuultavat pikselit haalistuvat pahimmillaan täysin väärän näköisiksi.

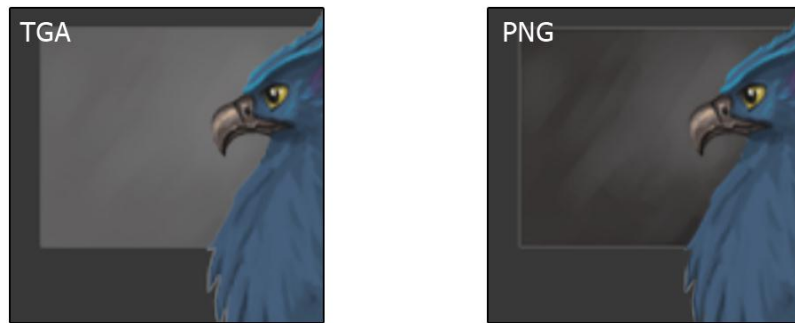
Tämän voi osittain korjata tekemällä tekstuurin alle jonkin sopivan värisen taustalayerin. Photoshopista katsoen kuvasta tulee siis täysin läpinäkymätön, mutta eri tiedostomuotoon tallennettaessa aiemmin määritelty alpha-kanava tallentuu oikein. Useimmiten yksivärinen tausta kuitenkin tekee joistain kuvan alueista hieman väärän värisiä. Yksi tapa saada kuvan RGB-kanavat tallentumaan varmasti oikean värisenä, on käyttää Flamingpearin *Solidify*-filtteriä. Solidify muuttaa kaikki pikselit läpinäkymättömiksi ja venyttää vielä värialueita niiden normaalien rajojen yli, joten alpha-kanavan reunat näkyvät oikean värisenä (Kuva 14).



Kuva 11. Testitekstuuri Photoshopissa katseltuna ja oikealla sen alpha-kanava.



Kuva 12. Ylläolevan kuvan RGB-kanavat TGA ja PNG-formaattiin tallentamisen jälkeen. TGA-kuvassa taustapalkin väri on sekoittunut valkoiseen taustaan. PNG-kuvassakin on omat outoutensa.



Kuva 13. Samat tekstuurit 3D-ohjelmassa katsottuna. TGA-kuvan läpikuultavat alueet ovat haalistuneet pilalle. Molempien kuvien reuna-alueilla on vaaleaa väriä.



Kuva 14. Vasemmalla Flamingpear Solidify-filteerillä käsitelty tekstuuri, joka on tallennettu TGA-muotoon alpha-kanavan kanssa. Oikealla sama tekstuuri 3D-ohjelmassa. Kaikki aiempien esimerkkien ongelmat ovat tiessään.

Kun kuva on tallennettu TGA-muotoon Solidify-käsittelyn jälkeen, voi tietysti kysyä kannattaako sitä enää muuttaa PNG-tiedostoksi. Itselläni suurin osa pelin tekstuureista on jo PNG-muodossa sen takia, että alpha-kanavan tallentamiseen liittyvät ongelmat ovat suhteellisen harvinaisia ja PNG on tämän takia nopeampi käyttää. TGA-tiedostot muutan PNG-formaattiin vain sen takia, että tällöin tiedostoja on helpompi organisoida.

TGA-kuvan saa muutettua oikein PNG-muotoon esimerkiksi *Pixelformer*-ohjelmalla ja pakkausta saa edelleen parannettua käyttämällä Ken Silvermanin *PNGOUT*-ohjelmaa. Kuvien Alpha-kanavan voisi tietysti tallentaa kokonaan erilliseen tiedostoon. Tällöin tiedostomäärän kasvu kuitenkin haittaisi työskentelyn selkeyttä.

6.6.4 DDS-tiedostot

Pelialalla tekstuurit tallennetaan useimmiten DDS-muotoisina. Tämän tyyppisissä peleissä DDS-formaattiin tallentaminen ei kuitenkaan kannata. Esimerkiksi Kulta-Jaskassa ja Unelmalomassa MIP mapit on kytketty kokonaan pois päältä, sillä niille ei ole tarvetta, kun tekstuurit piirretään ruudulle useimmiten 1:1-suhteessa. DDS-formaatin mahdollisuus kuvan tallennusvaiheessa luotaviin MIP mappeihin jäisi siis hyödyntämättä.

Pelit perustuvat useimmiten tarkasti käsin maalattuun grafiikkaan ja häviöllinen DXT-pakkaus saattaa heikentää kuvanlaatua selvästi. Sama laadun heikkeneminen on ilmeistä myös värikanavien tarkkuutta vähentäessä. Näin ollen kuvan tallentaminen DDS-formaattiin tarkoittaisi, ettei sen hyviä puolia voisi käyttää hyväksi ja kuvan joutuisi tallentamaan asetuksilla, jotka vastaisivat PNG-formaattia, lukuun ottamatta tämän tehokkaampaa, häviötöntä pakkausta.

Pääasiallinen syy jättää käyttämättä DXT-formaattia oli se, että PNG-formaatti on ollut talossa pitkään yleisessä käytössä, eikä DXT-formaatin tarkempaan tutkimiseen ole vielä ollut aikaa. Taustakuissa kuvalaadun huononeminen vaikuttaisi olevan DXT-formaatissa liian voimakasta. Joissain grafiikoissa sen käytöstä voisi kuitenkin olla hyötyä ilman laadun merkittävää heikkenemistä.



Kuva 15. Lähikuvia eri DXT-pakkaustyyppien vaikutuksesta testitekstuurin kuvanlaatuun.

6.6.5 Tekstuurien koko

Pelimoottorit vaativat yleensä että tekstuurien mitat seuraavat kahden potenssin sääntöä. Kuvan pituus- ja leveysmittojen pitäisi siis seurata kaavaa 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 jne. Usein kuvan pituus ja leveys voivat olla erimittaisia, mutta kummankin pitäisi seurata edellä mainittua kaavaa. Kannattaa huomioida että esimerkiksi luvut 384 ja 768 poikkeavat säännöstä vaikka näyttävätkin siihen hämmentävän sopivilta. Sääntö on peruja vanhojen grafiikkamoottoreiden tavasta käsitellä tekstureita. (Ahearn 2009.)

Siitä huolimatta että nykyisellä tekniikalla ei ole perusteita pakottautua käyttämään kahden potenssissa olevia tekstureita, on käytäntö vieläkin pelialalla yleinen. Esimerkiksi UDK ja Unity eivät suostu käsittelemään tekstureja, jotka eivät seuraa sääntöä. Säännön seuraaminen helpottaa tekstuurien asettelua tasaisiin ruudukoihin, mistä on usein iloa. Lisäksi voidaan varmistua, että tekstuurit toimivat myös vanhemmilla näytönohjaimilla tai joillain mobiililaitteilla.

Ongelmaa voi kiertää käyttämällä Atlas-tekstureja, eli kokoamalla useita tekstureita yhteen suureen tekstuuriin. Hyötyä voi saada jo pienistä kokonaisuuksista. Vaikkapa 1024x768-kokoisen taustakuvan voisi tallentaa 1024x1024 kokoiseen tekstuuriin ja käyttää ylimääräisen 1024x256-pikselin alueen animoitujen taustaelementtien tallentamiseen.

Kulta-Jaska ja Unelmaloma käyttävät molemmat useita tekstureita, jotka eivät seuraa kahden potenssin sääntöä. Pelit toimivat kyllä moitteetta, mutta alaa ajatellen voisi olla hyvää harjoitusta tehdä jatkossa tekstuurit sääntöä seuraten.

6.7 Työtiedostojen organisointi

Tiedostoja organisoidessa rupeaa helposti ajattelemaan liikaa ja kehittämään järkyttäviä, polveilevia hakemistosokkeloita. Hakemistorakenne on jo sinällään jäännös PC-tietokoneiden esihistoriasta, joten ei ole ihme, että sen tehokas käyttäminen on nykyään vaikeaa. Tiedostojen etsiminen insinöörimäisesti kategorisoiduista kansiohelvetistä on yleensä aikaa vievää ja turhauttavaa. Varsinkin jos kahlaa ensin kymmenen kansion läpi haluamaansa paikkaan ja sulkee sitten vahingossa ikkunan. Pahimmassa tapauk-

sessä tiedostoja tulee tallennettua väärään paikkaan yksinkertaisesti siksi, että haluaa välttää kuumottavat kansioseikkailut.

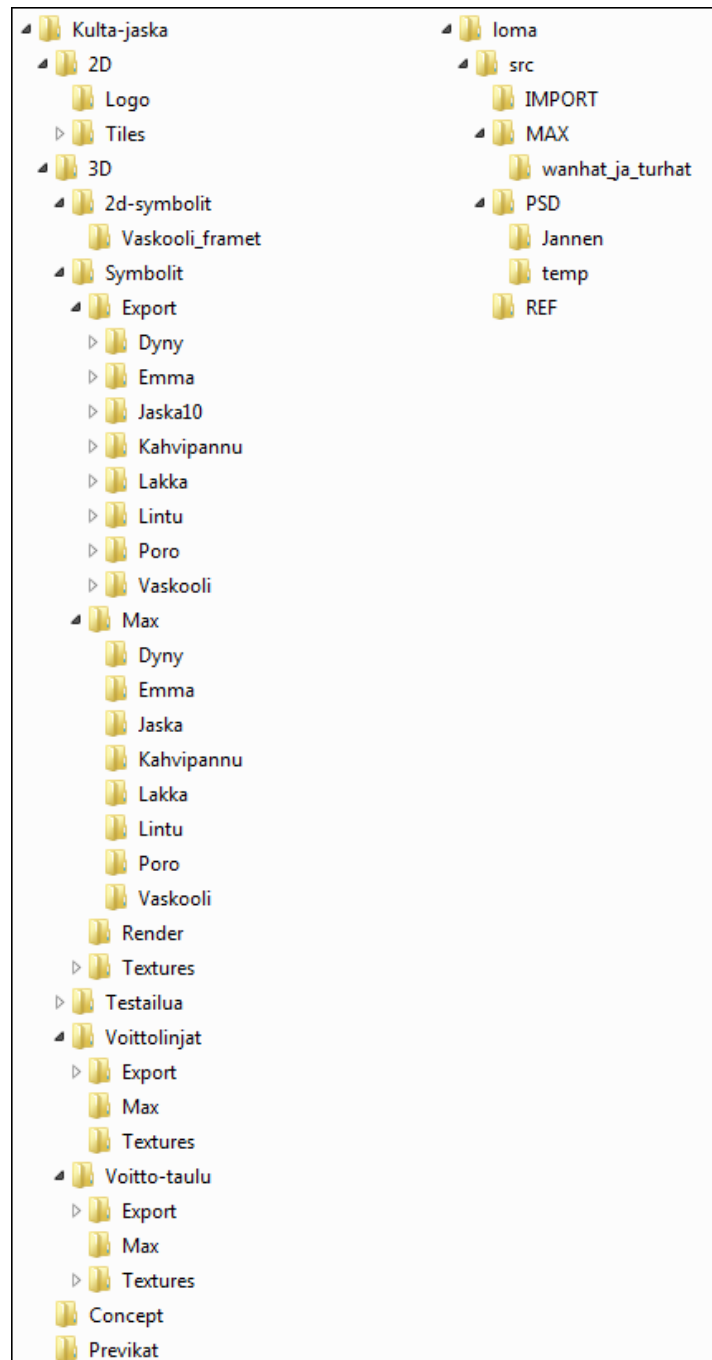
Oman kokemuksen mukaan ihannetapaus on, kun kaikki alahakemistot löytyvät heti päähakemiston alta. Poikkeuksia on tietysti aina ja alahakemistoja joutuu yleensä lisäämään viimeistään silloin jos graafikoita on useampia.

Useimmiten tiedostoista löytää helposti etsimänsä jo järjestelemällä näkymän tiedostotyyppin mukaan. Asioita helpottaa edelleen, jos käyttää selkeitä etu- ja jälkiliitteitä.

Tiedostojen sisällä layereiden ja objektien nimeäminen on yhtäläillä syytä tehdä huolellia. Ohjelmoijat viittaavat yleensä koodissa suoraan graafikon nimeämiin objekteihin. Välillä osa objekteista kierrätetään seuraavassa projektissa ja on tärkeää tietää, mitkä elementit ovat edellisestä pelistä. Photoshop tiedostojen suhteen on todennäköistä että muutkin päätyvät käsittelemään alkuperäisiä työtiedostoja, vaikka graafikoita olisi vain yksi ja huonosti nimetyt layerit hidastavat muiden työskentelyä.

IMPORT	loma_figure_alpit_tausta	loma_linjanapit	loma_vt_bonus_kehys
MAX	loma_figure_bonus	loma_linjanapit_off	loma_vt_bonus_kehys_digsy
PSD	loma_figure_bonus_2d	loma_linjanapit_on	loma_vt_bonus_voittosummat_1
REF	loma_figure_kapteeni	loma_linjanappi_tausta	loma_vt_bonus_voittosummat_2
linjanapit_off	loma_figure_kapteeni_2d	loma_lohku_liekki3_64x64	loma_vt_bonus_voittosummat_3
linjanapit_on	loma_figure_kenguru	loma_napit_150x100_eng	loma_vt_bonus_voittosummat_4
loma_acropolis_diffuse_384x384	loma_figure_kenguru_2d	loma_napit_150x100_fin	loma_vt_bonus_voittosummat_5
loma_bonus_aurinko_256x256	loma_figure_kenguru_maisema	loma_napit_150x100_swe	loma_vt_bonus_voittosummat_6
loma_bonus_fig_egypt	loma_figure_kiina_464x256	loma_ohjeinfo_150x100	loma_vt_bonus_voittosummat_7
loma_bonus_kortti_200x256	loma_figure_lentoemanta	loma_pelivalikko_150x100_eng	loma_vt_bonus_voittosummat_8
loma_bonus_kortti_glow_200x256	loma_figure_lentoemanta_2d	loma_pelivalikko_150x100_fin	loma_vt_bonus_kuvio_fin
loma_bonus_maa_256	loma_figure_lentokone_512x256	loma_pelivalikko_150x100_swe	loma_vt_kehys
loma_bonus_maisemat_1024x200	loma_figure_matkalaukku	loma_tajmahal_diffuse_384x384	loma_vt_kehys_digsy
loma_bonus_taivas_1024x384	loma_figure_matkalaukku_2d	loma_tausta_1024x768	loma_vt_tausta
loma_bonus_teksti_320x128	loma_figure_newyork_2d	loma_tulivuori_savu_128x128	loma_vt_tausta_digsy
loma_bonus_ylanaytot_1056x78	loma_figure_newyork_day_530x256	loma_vapaapelit_numerot_128x128_3	loma_vt_voittotahajallaan_fin
loma_easter_diffuse_768x254	loma_figure_newyork_night_530x256	loma_vapaapelit_numerot_128x128_10	loma_vt_voittosummat_01
loma_egypt_diffuse_686x256	loma_figure_newyork_other_302x260	loma_vapaapelit_numerot_128x128_20	loma_vt_voittosummat_02
loma_esb_diffuse_512x512	loma_figure_ranta	loma_vapaapelit_tekstit_512x128_eng	loma_vt_voittosummat_03
loma_figure_1	loma_figure_ranta_2d	loma_vapaapelit_tekstit_512x128_fin	loma_vt_voittosummat_04
loma_figure_5	loma_figure_ranta_render_alpha	loma_vapaapelit_tekstit_512x128_swe	loma_vt_voittosummat_05
loma_figure_alpit	loma_figure_venetsia	loma_vapaapelit_viiri_720x148	
loma_figure_alpit_2d	loma_figure_venetsia_2d	loma_voittolinjat	
loma_figure_alpit_393x256	loma_lentokone3d_diffuse_256x256	loma_volcano_diffuse_512x512	

Kuva 1. Unelmaloman grafiikoiden päähakemiston tiedostojen nimeäminen ei ole täydellistä, mutta suuri parannus Kulta-Jaskan vastaavasta. Alihakemistoihin piilotettuna asioita ei löytäisi ainakaan nopeammin



Kuva 2. Kulta-Jaskan ja sen jälkeen tehdyn Unelmaloman grafiikoiden projektihakemistojen rakenteet vierekkäin. Kulta-Jaskan tiedostojen hallinta oli huomattavasti hankalam-paa polveilevan hakemistorakenteen ja huonon nimeämislogiikan takia.

7 Yhteenveto ja lopputuloksen arviointia

7.1 Valmiiden tuotteiden arviointi

Kulta-Jaska ja Unelmaloma alkavat olla tarpeeksi vanhoja, että niitä pystyy jo katsomaan objektiivisella silmällä. Peliin teon ja julkaisun aikana olin melkoisen tyytymättömän lopputulokseen. Nyt pelejä pelatessa huomaa, että ne ovat oikeastaan ihan viihdyttäviä ja ovat toteutukseltaan jopa laadukkaampia, kuin monet kilpailevat tuotteet. Grafiikka ei ole omiin päämääriini ja esikuviini nähden kovin kummoista, mutta ajaa asiansa, eikä yleensä mene sieltä missä aita on matalin. Näin myöhemmin pystyy myös paremmin arvostamaan sitä, miten paljon muun tiimin suunnittelupanos näkyy myös graafisessa lopputuloksessa. Peleissä ja niiden grafiikassa on vielä paljon parantamisen varaa, mutta se tarkoittaa sitä, että työskentely pysyy mielenkiintoisena uusien pelien parissa.

7.2 Opinnäytetyöstä saavutettu hyöty

Opinnäytetyöhön keskittyminen oli arvokas tilaisuus arvioida omia työtapoja. Lopulta päädyin kyseenalaistamaan monia ratkaisujani ja löysin monia virheellisiä käsityksiä, mitä minulla oli erityisesti reaaliaikaisesta 3D-grafiikasta. Valmiin tuotteen tekeminen on minusta jo sinänsä arvokasta, mutta se aiheuttaa välillä ansaitsematonta omien työtapojen arvostusta. Kun työtavat joutui tällä tavoin esittelemään julkisesti, alkoi yhä suurempi osa oletetuista tiedoistani tuntua pelkiltä luuloilta. Nyt tekstin kirjoittamisen ja tutkimustyön jälkeen pystyn perustelemaan työtapani entistä paremmin ja toisaalta tiedän myös tapoja tehdä asiat paremmin jatkossa. Lisäksi löysin paljon hyödyllistä materiaalia, jota en vielä ehtinyt edes kokonaan opiskella.

Varsinaisista työtapojen esittelyistä olisi ollut ainakin itselleni iloa, silloin kun aikoinaan pohdin Kulta-Jaskan mahdollisia toteutustapoja. Tekniikat eivät ole erityisen monimutkaisia, mutta kenties niistä on hyötyä myös muille. Ehkäpä pelialaa ja rahapelejä tuntematon saa myös tekstistä paremman kuvan siitä, mitä asioita peligraafikon työhön kuuluu.

7.3 Työskentely tulevaisuudessa

Toistaiseksi pelien grafiikkaa tehdessä on joutunut keskittymään paljon teknisten ratkaisujen pohtimiseen. Tässä opinnäytetyössä esiteltyt työtavat alkavat kuitenkin viimein olla rutiininomaisia ja olen suorastaan innoissani tekniikan mahdollisuuksista. Kun tekniset ongelmat vievät yhä vähemmän aikaa, on mahdollisuus keskittyä enemmän sisältöön.

Saattaa kuitenkin olla, että nämä tekniikat ovat liian rajallisia esimerkiksi esirenderöityihin animaatioihin verrattuna. Viimeaikoina pelikohtaiset tiimikoot ovat osoittaneet kasvamisen merkkejä, joten saattaa olla, että kohta joutuu jälleen opettelemaan uusia työtapoja, kun vaativamman grafiikan tekeminen muuttuu mahdolliseksi. Työskentelyyn rutinoitumisessa on toki muitakin osa-alueita, kuin tekninen puoli. Siinä mielessä uusien tekniikoiden omaksuminen lienee helpompaa kuin aiemmin.

Kirjalähteet

Ahearn, Luke (2009). 3D Game Textures – Second Edition. Elsevier. Burlington.

Verkkolähteet

Linne, Riccard (2005).

Making Quality Game Textures. Gamasutra. (23.11.2005).

<http://www.gamasutra.com/view/feature/2467/making_quality_game_textures.php>.

(Luettu 23.5.2011).

Refenes, Tommy (2009).

Super Meat Boy development blog. (27.6.2009).

<<http://supermeatboy.blogspot.com/2009/06/i-made-something-neato.html>>.

(Luettu 24.4.2011).

Roelofs, Greg (2011).

PNG (Portable Network Graphics) Home Site. (6.1.2011).

<<http://www.libpng.org/pub/png/>>.

(Luettu 24.5.2011).

Schwaber, Ken, Sutherland, Jeff (2010). Scrum Guide. (helmikuu 2010).

<<http://www.scrum.org/scrumguides/>>.

(Luettu 4.6.2010).

The Spriters Resource (2009).

PSX / PS2 > O > Odin Sphere > Velvet (Pooka). (1.5.2009).

<http://www.spriters-resource.com/psx_ps2/odinsphere/sheet/22752>.

(Luettu 24.4.2011).

Wikipedia (2011).

Truevision TGA. (12.1.2011).

<http://en.wikipedia.org/wiki/Truevision_TGA>.

(Luettu 24.5.2011).

Wloka, Matthias (2005).

Batching 4EVA. (7.3.2005).

<http://http.download.nvidia.com/developer/presentations/2005/GDC/Direct3D_Day/D3DTutorial03_Optimization.pdf>.

(Luettu 27.2.2011).

Kuvalähteet

Kuva 4: Muramasa: The Demon Blade on oiva todiste 2D-grafiikan elinvoimaisuudesta.

<<http://www.pocketgamer.co.uk/r/Various/The+Rumour+Mine/feature.asp?c=23584>>